Estimators, Loss Functions and Decision Theory

Prof. Nicholas Zabaras
Center for Informatics and Computational Science
https://cics.nd.edu/
University of Notre Dame
Notre Dame, Indiana, USA

Email: nzabaras@gmail.com
URL: https://www.zabaras.com/

January 30, 2019

Contents

Following closely

- Chris Bishops' PRML book, Chapter 1
- Kevin Murphy's Machine Learning: A probabilistic Perspective, Chapter 5

Bayesian Decision Theory

- Beyond Bayesian inference, a goal is to convert our beliefs into optimal actions.
- State a statistical decision problem as a game against nature.
  - nature picks a state or parameter, $y \in \mathcal{Y}$, unknown to us,
  - generates an observation, $\boldsymbol{x} \in \mathcal{X}$, which we get to see.
  - We then have to make a decision, that is, we have to choose an action `a' from some action space $\mathcal{A}$.
- Finally we incur some loss, $L(y, a)$, which measures how compatible our action 'a' is with nature's hidden state y.
- We might use misclassification loss $L(y, a) = \mathbb{I}(y \neq a)$, or squared loss, $L(y, a) = (y - a)^2$.

Bayes Estimator

- Devise a decision procedure $\delta : \mathcal{X} \rightarrow \mathcal{A}$ which specifies the optimal action for each possible input. Minimize the expected loss:

$$\delta(x) = \arg\min_{a \in \mathcal{A}} \mathbb{E}\left[L(y,a)\right]$$

- In economics, we talk of a utility function $U(y,a) = -L(y,a)$.

$$\delta(x) = \arg\max_{a \in \mathcal{A}} \mathbb{E}\left[U(y,a)\right]$$

- This is called the maximum expected utility principle.
- In the Bayesian approach, the optimal action, having observed $\boldsymbol{x}$, is defined as *the action `a' that minimizes the posterior expected loss:*

$$\rho(a \mid \boldsymbol{x}) = \mathbb{E}_{p(y|\boldsymbol{x})}\left[L(y,a)\right] = \sum_y L(y,a)\, p(y \mid \boldsymbol{x})$$

- Hence the Bayes estimator, also called the Bayes decision rule, is given by

$$\delta(\boldsymbol{x}) = \arg\min_{a \in \mathcal{A}} \rho(a \mid \boldsymbol{x})$$

MAP Estimate Minimizes 0-1 Loss

- The $0-1$ Loss is defined as:

$$L(y,a)=\mathbb{I}\left(y\neq a\right)=\begin{cases}0\ if\ \ a=y\\1\ if\ \ a\neq y\end{cases}$$

- This is commonly used in classification problems where y is the true class label and $a=\hat{y}$ the estimate.

- The posterior expected loss is then:

$$\rho(a\,|\,\boldsymbol{x})=\int\mathbb{I}\left(y\neq a\right)p(y\,|\,\boldsymbol{x})dy=\int\left(1-\mathbb{I}\left(y=a\right)\right)p(y\,|\,\boldsymbol{x})dy=1-p(y=a\,|\,\boldsymbol{x})$$

- Hence the action that minimizes the expected loss is the posterior mode or MAP estimate

$$\hat{y}(\boldsymbol{x})=\underset{y\in\mathcal{Y}}{\operatorname{argmax}}p(y|\boldsymbol{x})$$

Hypothesis Testing: 0-1 Loss

- Given $\boldsymbol{x}$, you are trying to guess if $y \in \mathcal{Y}_0$. With $a = 1$ (you guess $y \in \mathcal{Y}_0$) or $a = 0$ (you guess $y \notin \mathcal{Y}_0$) define the loss function

$$L(y,a) = \begin{cases} 1-\alpha & if \;\; y \in \mathcal{Y}_0 \\ \alpha & otherwise \end{cases}$$

- The posterior expected loss is given as follows:

$$\mathbb{E}^{\pi}\left[L(y,a) \mid \boldsymbol{x}\right] = \Pr\left(y \in \mathcal{Y}_0 \mid \boldsymbol{x}\right)(1-a) + \Pr\left(y \notin \mathcal{Y}_0 \mid \boldsymbol{x}\right)a = \begin{cases} \Pr\left(y \in \mathcal{Y}_0 \mid \boldsymbol{x}\right) & if \;\; a = 0 \\ \Pr\left(y \notin \mathcal{Y}_0 \mid \boldsymbol{x}\right) & if \;\; a = 1 \end{cases}$$

- Our estimator will be $a = 0$ when:

$$P^{\pi}(y \in \mathcal{Y}_0 \mid \boldsymbol{x}) < P^{\pi}(y \notin \mathcal{Y}_0 \mid \boldsymbol{x}) = 1 - P^{\pi}(y \in \mathcal{Y}_0 \mid \boldsymbol{x}) \Rightarrow P^{\pi}(y \in \mathcal{Y}_0 \mid \boldsymbol{x}) < \frac{1}{2}$$

- The associated Bayes estimate minimizing the posterior expected loss is then:

$$\delta^{\pi}(\boldsymbol{x}) = \begin{cases} 1 & if \;\; \Pr\left(y \in \mathcal{Y}_0 \mid \boldsymbol{x}\right) > \frac{1}{2} \\ 0 & otherwise \end{cases}$$

Hypothesis Testing: α_0-α_1 Loss

- Consider the weighted $0-1$ (or a_0-a_1) loss extension

$$L(y,a)=\begin{cases}(1-a)\alpha_0 \ if \ y\in\mathcal{Y}_0\\ a\alpha_1 \quad otherwise\end{cases}=\begin{cases}0 & if \ a=\mathbb{I}_{\mathcal{Y}_0}(y)\ (otherwise)\\ a_0 & if \ y\in\mathcal{Y}_0 \ and \ a=0\\ a_1 & if \ y\in\mathcal{Y}_1 \ and \ a=1\end{cases}$$

- The posterior expected loss is given as follows:

$$\mathbb{E}^{\pi}\left[L(\theta,\alpha)\,|\,\boldsymbol{x}\right]=\alpha_0\Pr\left(y\in\mathcal{Y}_0\,|\,\boldsymbol{x}\right)(1-\alpha)+\alpha_1\Pr\left(y\notin\mathcal{Y}_0\,|\,\boldsymbol{x}\right)\alpha==\begin{cases}\alpha_0\Pr\left(y\in\mathcal{Y}_0\,|\,\boldsymbol{x}\right) & if \ a=0\\ \alpha_1\Pr\left(y\notin\mathcal{Y}_0\,|\,\boldsymbol{x}\right) & if \ a=1\end{cases}$$

- Our Bayes estimator will be $a=0$ if

$$a_0P^{\pi}(y\in\mathcal{Y}_0|\boldsymbol{x})<a_1P^{\pi}(y\notin\mathcal{Y}_0\,|\,\boldsymbol{x})=a_1\left(1-P^{\pi}(y\in\mathcal{Y}_0\,|\,\boldsymbol{x})\right)\Rightarrow$$

$$P^{\pi}(y\in\mathcal{Y}_0\,|\,\boldsymbol{x})<\frac{a_1}{a_0+a_1}$$

i.e.

$$\delta^{\pi}(\boldsymbol{x})=\begin{cases}1 & if \ P^{\pi}(y\in\mathcal{Y}_0|\boldsymbol{x})>a_1/(a_0+a_1)\\ 0 & otherwise\end{cases}$$

The Posterior Mean Minimizes the L_2 Loss

$$\mathbb{E}\left[L_2\right] = \int \left(y - a\right)^2 \, p(y \mid \boldsymbol{x}) dy = \mathbb{E}\left[y^2 \mid \boldsymbol{x}\right] - 2a\mathbb{E}\left[y \mid \boldsymbol{x}\right] + a^2$$

- Setting the first derivative with respect to `a' equal to 0:

$$\hat{y} = \mathbb{E}[y|\boldsymbol{x}] = \int y p(y|\boldsymbol{x}) dy$$

- This is often called the minimum mean squared error estimate or MMSE.

The L_2 (Quadratic) Loss

- Historically, this is the first loss function

$$L(y,a) = (y-a)^2$$

- The Bayes estimate $\delta^{\pi}(x)$ associated with the prior π and with the quadratic loss is the posterior expectation

$$\delta^{\pi}(\boldsymbol{x}) = \mathbb{E}^{\pi}[y \mid \boldsymbol{x}] = \frac{\int_{\mathcal{Y}} y f(\boldsymbol{x} \mid y)\pi(y)dy}{\int_{\mathcal{Y}} f(\boldsymbol{x} \mid y)\pi(y)dy}$$

- For another proof of this note that:

$$\mathbb{E}^{\pi}\left[L(y,a) \mid \boldsymbol{x}\right] = \mathbb{E}\left[(y-a)^T(y-a) \mid \boldsymbol{x}\right] = \mathbb{E}\left[y^2 \mid \boldsymbol{x}\right] - 2a^T\mathbb{E}\left[y \mid \boldsymbol{x}\right] + \|a\|^2 =$$

$$= \mathbb{E}\left[y^2 \mid \boldsymbol{x}\right] - \mathbb{E}\left[y \mid \boldsymbol{x}\right]^2 + (\mathbb{E}\left[y \mid \boldsymbol{x}\right] - a)^2$$

- The minimizer of the quadratic loss is then obtained when: $a = \mathbb{E}[y \mid \boldsymbol{x}]$

The Absolute Error Loss

- The absolute error loss is defined as:

$$L(y,a)=|y-a| \qquad \text{or} \qquad L_{k_1,k_2}(y,a)=\begin{cases} k_2(y-a) & if\ \ y>a \\ k_1(a-y)\ otherwise \end{cases}$$

- The posterior expected loss is (see CP Roberts, The Bayesian Choice):

$$\mathbb{E}^{\pi}\left[L_{k_1,k_2}(y,a)\,|\,\boldsymbol{x}\right]=\int_{-\infty}^{a}k_1(a-y)\pi(y\,|\,\boldsymbol{x})dy+\int_{a}^{\infty}k_2(y-a)\pi(y\,|\,\boldsymbol{x})dy=\int_{-\infty}^{a}k_1(a-\theta)\pi(\theta\,|\,\boldsymbol{x})d\theta$$

$$+\int_{a}^{\infty}k_2(\theta-a)\pi(\theta\,|\,\boldsymbol{x})d\theta=\int_{-\infty}^{a}k_1(a-\theta)\frac{d\Pr(y<\theta\,|\,\boldsymbol{x})}{d\theta}d\theta+\int_{a}^{\infty}k_2(\theta-a)\frac{d\Pr(y>\theta\,|\,\boldsymbol{x})}{d\theta}d\theta=\text{(integration by parts)}$$

$$\int_{-\infty}^{a}k_1\Pr(y<\theta\,|\,\boldsymbol{x})d\theta+\int_{a}^{\infty}k_2\Pr(y>\theta\,|\,\boldsymbol{x})d\theta$$

- Taking derivative wrt to a gives $k_1\Pr(y<a\,|\,\boldsymbol{x})-k_2\Pr(y>a\,|\,\boldsymbol{x})=0\Rightarrow$

$$k_1\Pr(y<a\,|\,\boldsymbol{x})-k_2(1-\Pr(y<a\,|\,\boldsymbol{x}))=0\Rightarrow \quad \Pr(y<\hat{y}|\boldsymbol{x})\equiv\Pr(y<\mathrm{a}|\boldsymbol{x})=\frac{k_2}{k_1+k_2}$$

- The associated Bayes estimate is $k_2/(k_1+k_2)$ fractile of $\pi(y|\boldsymbol{x})$. For $k_2=k_1=1$, this becomes the posterior median.

MAP Estimator

- With no loss function, we use the maximum a posteriori (MAP) estimator

$$\arg\min_{y} \ell\left(y \middle| \boldsymbol{x}\right)\pi(y)$$

  - Penalized likelihood estimator
  - Further appeal in restricted parameter spaces

- As an example, consider: $f(x|y) = \frac{1}{\pi}\left[1+(x-y)^2\right]^{-1}$

and $\pi(y) = \frac{1}{2}e^{-|y|}$

Then the MAP estimate of y is always

$$\delta^{\pi}(x) = 0$$

MAP Estimator (Binomial Probability)

- Consider a binomial distribution $x|\theta \sim \mathcal{B}(n,\theta)$

Possible priors:

$$\pi^J(y) = \frac{1}{Beta(1/2,1/2)} y^{-1/2}(1-y)^{-1/2}$$

Binomial	$\theta \sim \text{Bin}(n,p)$ $p(\theta) = \text{Bin}(\theta\|n,p)$	'sample size' n (positive integer) 'probability' $p \in [0,1]$
$p(\theta) = \binom{n}{\theta} p^\theta (1-p)^{n-\theta}$ $\theta = 0,1,2,\ldots,n$		$\text{E}(\theta) = np$ $\text{var}(\theta) = np(1-p)$ $\text{mode}(\theta) = \lfloor (n+1)p \rfloor$

$$\pi_1(y) = 1 \; and \; \pi_2(y) = y^{-1}(1-y)^{-1}$$

- The posterior is a beta distribution. The corresponding MAP estimators (see here for the mode of the beta Distribution):

$$\delta^{\pi_J}(y) = \max\left(\frac{x-1/2}{n-1}, 0\right)$$

$$\delta^{\pi_1}(y) = x/n$$

$$\delta^{\pi_2}(y) = \max\left(\frac{x-1}{n-2}, 0\right)$$

Beta	$\theta \sim \text{Beta}(\alpha,\beta)$ $p(\theta) = \text{Beta}(\theta\|\alpha,\beta)$	'prior sample sizes' $\alpha > 0, \beta > 0$
$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$ $\theta \in [0,1]$		$\text{E}(\theta) = \frac{\alpha}{\alpha+\beta}$ $\text{var}(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$ $\text{mode}(\theta) = \frac{\alpha-1}{\alpha+\beta-2}$

Decision Theory: Loss Functions

- Purpose of Bayesian inference: to provide us with a decision $a \in \mathcal{A}$.
- Requires an evaluation criterion (loss function) for decisions and estimators

$$L(y, a)$$

- There exists an axiomatic derivation of the existence of a loss function.
- Decision procedure $\delta^{\pi} \equiv \hat{y}$ usually called estimator (while its value $\delta^{\pi}(\boldsymbol{x})$ is called estimate of y)
- Impossible to uniformly minimize (in $\mathcal{A}$) the loss function $L(y, a)$ when y is unknown.

C. P. Robert, *The Bayesian Choice*, *Springer, 2nd edition,* chapter 2 (full text available for Cornell students)

Bayes' Estimator

- Integrate over the space $\mathcal{Y}$ to get the **posterior expected loss**

$$\mathbb{E}^{\pi}\left[L(y,a)\middle|\boldsymbol{x}\right]=\int_{\mathcal{Y}}L(y,a)\pi(y|\boldsymbol{x})dy$$

and minimize with respect to a.

Bayes' Estimator

A Bayes estimate associated with a prior distribution π and a loss function L is

$$\arg\min_{a}\mathbb{E}^{\pi}\left[L(y,a)\middle|\boldsymbol{x}\right]=\arg\min_{a}\int_{\mathcal{Y}}L(y,a)\pi\left(y\mid\boldsymbol{x}\right)dy$$

Decision Theory for Regression

- Inference step

  Determine $p(\boldsymbol{x}, t)$

- Decision step

  For given $\boldsymbol{x}$, make optimal prediction, $y(\boldsymbol{x})$, for t at a loss $L(t, y(\boldsymbol{x}))$

- *The expected loss function:*

$$\mathbb{E}[L] = \int\int L(t, y(\boldsymbol{x}))\, p(\boldsymbol{x}, t)\, d\boldsymbol{x} dt$$

- A common choice of loss function in regression problems is the squared loss

$$\mathbb{E}[L] = \int\int \big(y(\boldsymbol{x}) - t\big)^2\, p(\boldsymbol{x}, t)\, d\boldsymbol{x} dt$$

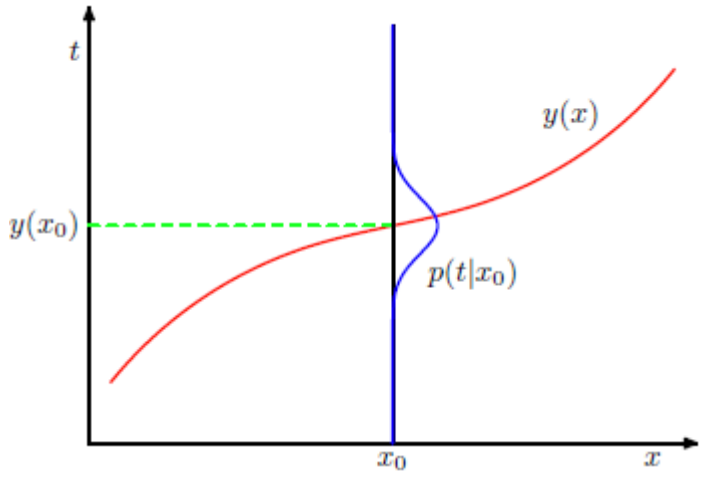
Squared Loss Function for Regression

$$\mathbb{E}[L] = \int\int \big(y(\boldsymbol{x}) - t\big)^2 \, p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

- Our goal is to choose $y(\boldsymbol{x})$ so as to minimize $\mathbb{E}[L]$. If we take no constraints on $y(\boldsymbol{x})$, we can use calculus of variations:

$$\frac{\delta \mathbb{E}[L]}{\delta y(\boldsymbol{x})} = 2\int \big(y(\boldsymbol{x}) - t\big) \, p(\boldsymbol{x}, t) dt \Rightarrow$$

$$y(\boldsymbol{x}) = \frac{\int t p(\boldsymbol{x}, t) dt}{p(\boldsymbol{x})} = \int t p(t \mid \boldsymbol{x}) dt = \underbrace{\mathbb{E}_t[t \mid \boldsymbol{x}]}_{\text{Regression function}}$$



Squared Loss Function for Regression

- We can also derive this result in a slightly different way, by expanding the square term as follows:

$$\big(y(\boldsymbol{x})-t\big)^2=\big(y(\boldsymbol{x})-\mathbb{E}[t\,|\,\boldsymbol{x}]+\mathbb{E}[t\,|\,\boldsymbol{x}]-t\big)^2=$$

$$=\big(y(\boldsymbol{x})-\mathbb{E}[t\,|\,\boldsymbol{x}]\big)^2+2\big(y(\boldsymbol{x})-\mathbb{E}[t\,|\,\boldsymbol{x}]\big)\big(\mathbb{E}[t\,|\,\boldsymbol{x}]-t\big)+\big(\mathbb{E}[t\,|\,\boldsymbol{x}]-t\big)^2$$

- Substituting into the loss function, the 2nd term gives:

$$2\int\big(y(\boldsymbol{x})-\mathbb{E}[t\,|\,\boldsymbol{x}]\big)\underbrace{\Big(\int\big(\mathbb{E}[t\,|\,\boldsymbol{x}]-t\big)\,p(t\,|\,\boldsymbol{x})dt\Big)}_{=0}p(\boldsymbol{x})d\boldsymbol{x}=0$$

- We are left with these two terms:

$$\mathbb{E}[L]=\int\big(y(\boldsymbol{x})-\mathbb{E}[t\,|\,\boldsymbol{x}]\big)^2\,p(\boldsymbol{x})d\boldsymbol{x}+\int\int\Big(\big(\mathbb{E}[t\,|\,\boldsymbol{x}]-t\big)^2\,p(t\,|\,\boldsymbol{x})dt\Big)p(\boldsymbol{x})d\boldsymbol{x}$$

$$\mathbb{E}[L]=\int\big(y(\boldsymbol{x})-\mathbb{E}[t\,|\,\boldsymbol{x}]\big)^2\,p(\boldsymbol{x})d\boldsymbol{x}+\int\mathrm{var}[t\,|\,\boldsymbol{x}]\,p(\boldsymbol{x})d\boldsymbol{x}$$

Squared Loss Function for Regression

$$\mathbb{E}[L] = \int \big(y(\boldsymbol{x}) - \mathbb{E}[t \mid \boldsymbol{x}]\big)^2 \, p(\boldsymbol{x}) d\boldsymbol{x} + \int \mathrm{var}[t \mid \boldsymbol{x}] \, p(\boldsymbol{x}) d\boldsymbol{x}$$

- Minimization of the 1st term gives the regression function.
- The 2nd term (the variance of the distribution of t averaged over $\boldsymbol{x}$) is an irreducible value of the loss function.

Note that the L_2 loss penalizes deviations from the truth quadratically and thus is sensitive to outliers. A more robust alternative is the L_1 loss.

Alternate Approaches to Regression

- As was the case with the general decision problems, we have three different approaches to regression:
  - Find $p(\boldsymbol{x}, t)$, then find $p(t|\boldsymbol{x}) = p(\boldsymbol{x}, t)/p(\boldsymbol{x})$ and finally find the solution $\mathbb{E}[t \mid \boldsymbol{x}]$
  - First solve for the conditional $p(t|\boldsymbol{x})$ and then find $\mathbb{E}[t \mid \boldsymbol{x}]$
  - Compute the regression function $y(\boldsymbol{x})$ directly from the data.

Posterior Expected Loss for Regression

- Consider a prediction function $y: \boldsymbol{x} \rightarrow t$ and suppose we have some cost function $L(t, y(\boldsymbol{x}))$ which gives the cost of predicting $y(\boldsymbol{x})$ when the truth is t. We can define the loss incurred as:

$$L(\theta, y(.)) = \mathbb{E}_{(\boldsymbol{x},t)\sim p(\boldsymbol{x},t|\theta)}\left[L\left(t, y(\boldsymbol{x})\right)\right] = \sum_{\boldsymbol{x}}\sum_{t} L\left(t, y(\boldsymbol{x})\right) p(\boldsymbol{x}, t \mid \theta)$$

- Here θ is the unknown state of nature (data generating mechanism).

- This is known as the *generalization error.* Our goal now is *to minimize the posterior expected loss* given by:

$$\rho(y(.) \mid \mathcal{D}) = \int p(\theta \mid \mathcal{D}) L(\theta, y(.)) d\theta$$

- *The solution is:* $\hat{y}(\boldsymbol{x}) = \int t p(t|\boldsymbol{x}, \mathcal{D}) dt$ (posterior mean).

The Posterior Mean Minimizes the L_2 Loss

- In linear regression we have:

$$p(t \mid \boldsymbol{x}, \boldsymbol{\theta}) = \mathcal{N}\left(t \mid \boldsymbol{x}^T \boldsymbol{\theta}, \sigma^2\right)$$

- Minimizing the posterior expected (quadratic) loss gives:

$$\hat{y}(\boldsymbol{x}) = \int \int t p(t|\boldsymbol{x}, \boldsymbol{\theta}) dt \, p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} = \int t p(t|\boldsymbol{x}, \mathcal{D}) dt = \mathbb{E}[t|\boldsymbol{x}, \mathcal{D}] = \boldsymbol{x}^T \mathbb{E}[\boldsymbol{\theta}|\mathcal{D}]$$

- In this case, the optimal estimate for given training data $\mathcal{D}$ is

$$\hat{y}(\boldsymbol{x}) = \mathbb{E}[t|\boldsymbol{x}, \mathcal{D}] = \boldsymbol{x}^T \mathbb{E}[\boldsymbol{\theta}|\mathcal{D}]$$

- Note that here we just plug-in the posterior mean parameter estimate. This is the optimal thing to do regardless of what prior we use for $\boldsymbol{\theta}$.

The Minkowski Loss Function

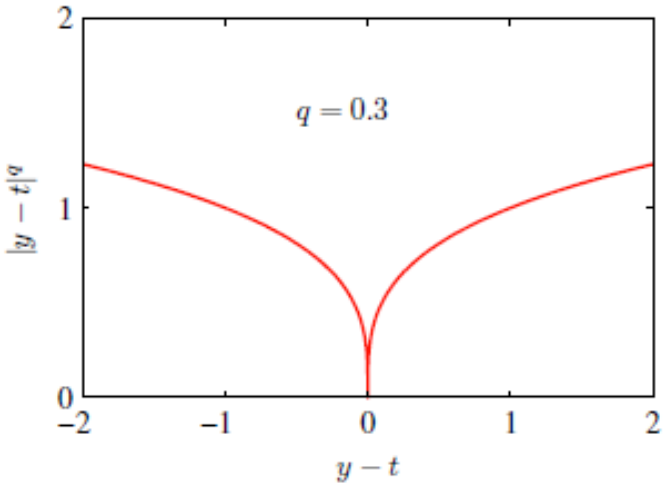
- There are situations in which squared loss can lead to very poor results.

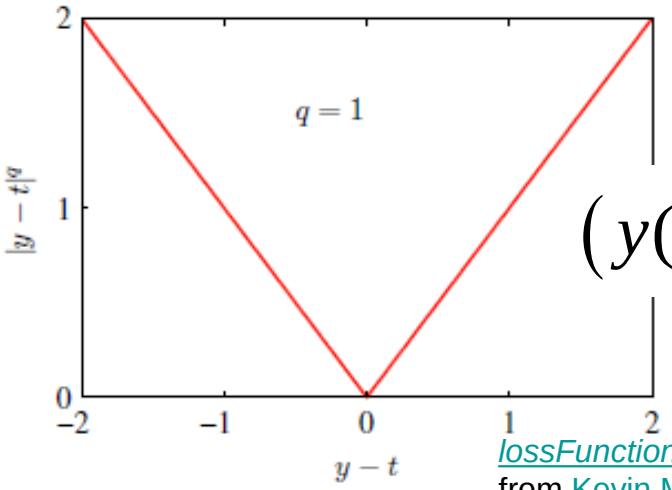
- A simple generalization of the squared loss is the Minkowski loss, whose expectation is given by

$$\mathbb{E}\left[L_q\right] = \int\int \left(y(\boldsymbol{x}) - t\right)^q \, p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

which reduces to the expected squared loss for $q = 2$.

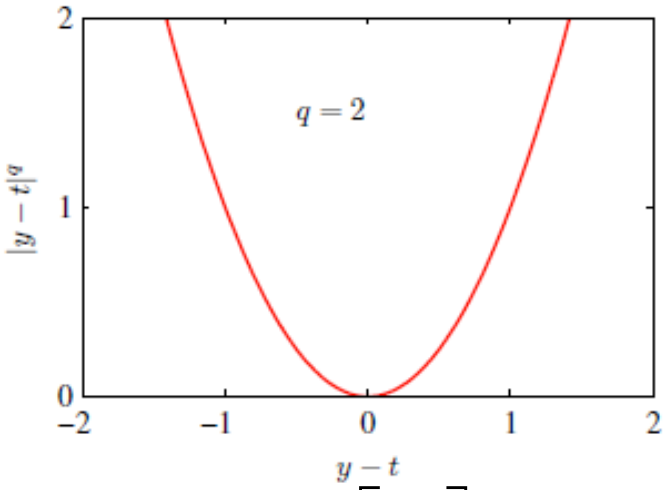
The Minkowski Loss Function

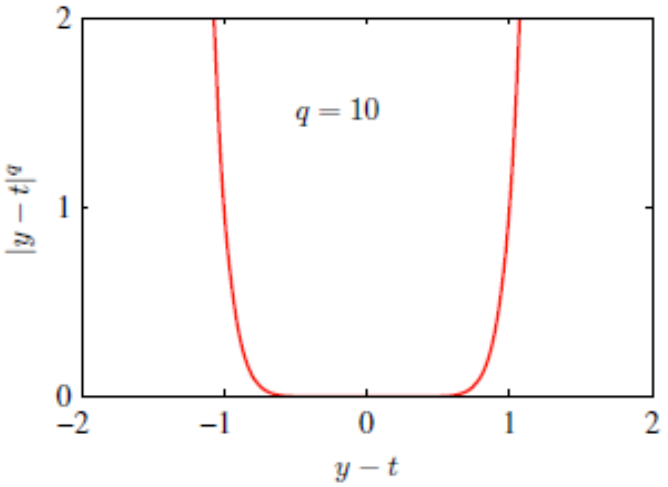




$$\left(y(\boldsymbol{x}) - t\right)^q$$

lossFunctionFig
from Kevin Murphys' PMTK





The minimum of $\mathbb{E}\left[L_q\right]$ is the conditional mean ($q = 2$), *the conditional median* ($q = 1$) *& the conditional mode for* $q \to 0$.

The Minkowski Loss Function

$$\mathbb{E}\left[L_q\right] = \int\int \left(y(\boldsymbol{x}) - t\right)^q \, p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

- We can *choose $y(\boldsymbol{x})$ independently for each $\boldsymbol{x}$* and thus we can minimize instead:

$$\int \left(y(\boldsymbol{x}) - t\right)^q \, p(t \mid \boldsymbol{x}) dt$$

- Setting the directional derivative wrt $y(\boldsymbol{x})$ equal to 0 gives:

$$q \int \left|y(\boldsymbol{x}) - t\right|^{q-1} sign\left(y(\boldsymbol{x}) - t\right) p(t \mid \boldsymbol{x}) dt$$

$$= q \int_{-\infty}^{y(\boldsymbol{x})} \left|y(\boldsymbol{x}) - t\right|^{q-1} p(t \mid \boldsymbol{x}) dt - q \int_{y(\boldsymbol{x})}^{+\infty} \left|y(\boldsymbol{x}) - t\right|^{q-1} p(t \mid \boldsymbol{x}) dt = 0$$

- For $q = 1$: $\int_{-\infty}^{y(\boldsymbol{x})} p(t \mid \boldsymbol{x}) dt = \int_{y(\boldsymbol{x})}^{+\infty} p(t \mid \boldsymbol{x}) dt = 0 \Rightarrow y(\boldsymbol{x})$ *conditional median*

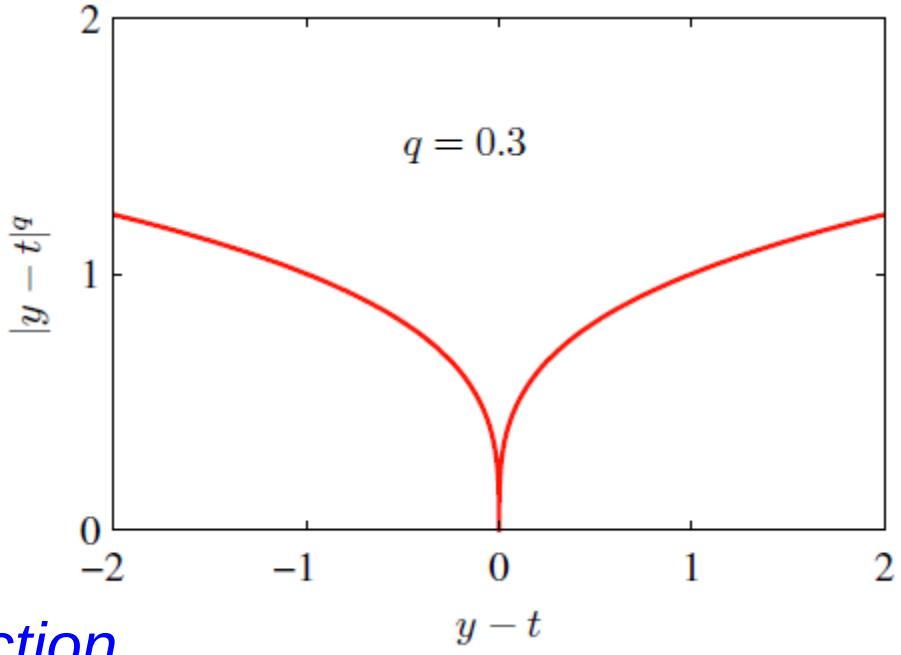
The Minkowski Loss Function

$$\mathbb{E}\left[L_q\right] = \int\int \left(y(\boldsymbol{x}) - t\right)^q \, p(\boldsymbol{x}, t) d\boldsymbol{x} dt$$

- For $q \to 0$, $\left(y(\boldsymbol{x}) - t\right)^q$ as a function of t is close to 1 except near $y(\boldsymbol{x}) = t$. Thus the value of

$$\int \left(y(\boldsymbol{x}) - t\right)^q \, p(t \mid \boldsymbol{x}) dt$$

will be close to 1 ($p(t|\boldsymbol{x})$ is normalized) and reduced only near the notch $t = y(\boldsymbol{x})$.



- Choose the location of the notch to coincide with the largest value of $p(t|\boldsymbol{x})$. *Thus the max reduction is for $y(\boldsymbol{x}) =$ conditional mode.*

Decision Theory for Classification

Consider a classification problem:

- One step solution: Train a function to decide the class
- Two step solution:
  - ✓ Inference: Infer posterior probabilities $p(C_k|\boldsymbol{x})$
  - ✓ Decision: Use posterior probabilities to decide the class labels

Decision Theory

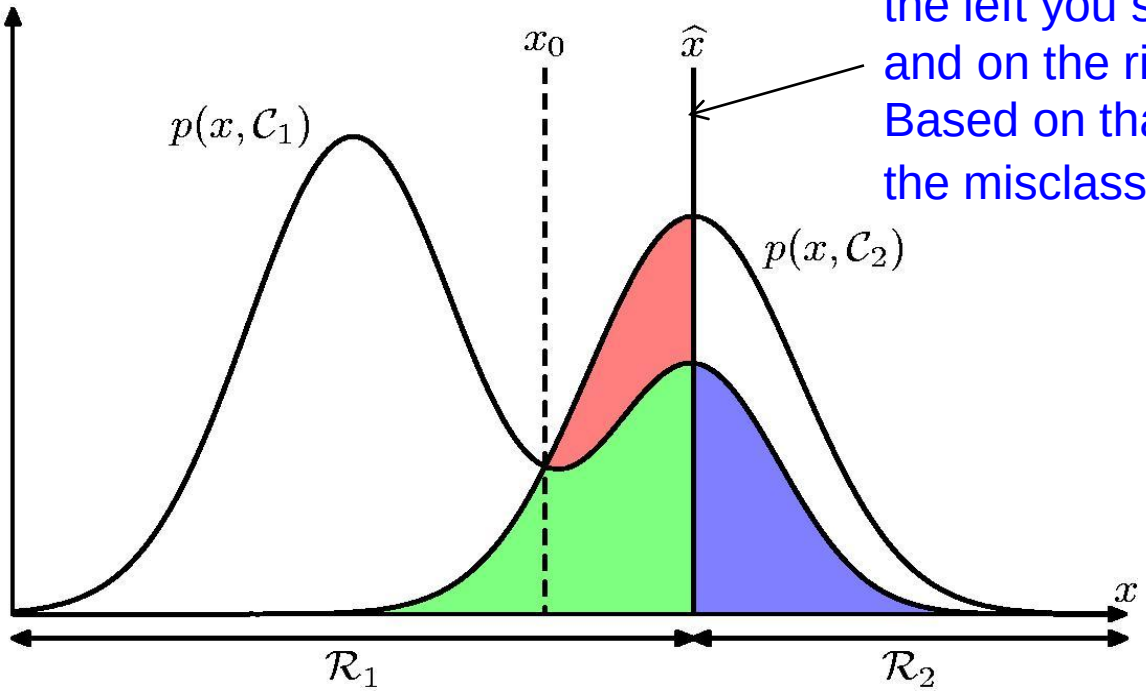
- Medical diagnosis problem: we have taken an X –ray image of a patient and wish to determine whether the patient has cancer or not.
- Inference problem: determining the joint $p(\boldsymbol{x}, C_k)$ which gives us the most complete probabilistic description.
- In the end we must decide either to give treatment to the patient or not, and we would like this choice to be optimal in some sense.
- Decision Step -- *how to make optimal decisions given the appropriate probabilities (Duda et al., Berger, Bather)*

- Duda, R. O. and P. E. Hart (1973). *Pattern Classification and Scene Analysis*. Wiley
- Berger, J. O. (1985). *Statistical Decision Theory and Bayesian Analysis* (Second ed.). Springer.
- Bather, J. (2000). *Decision Theory: An Introduction to Dynamic Programming and Sequential Decisions*. Wiley.

Minimizing the Misclassification Rate

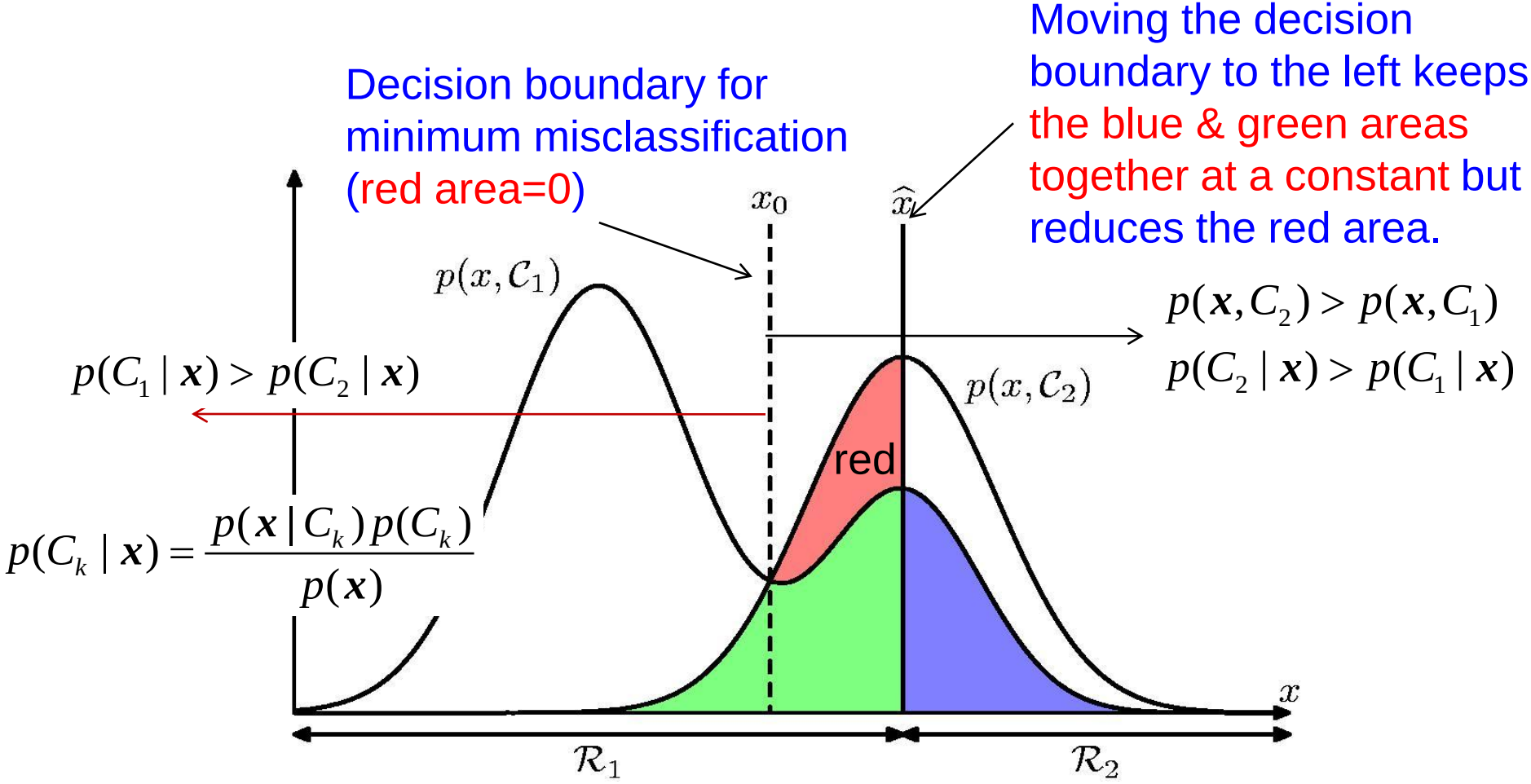
- A mistake occurs when an input vector belonging to class C_1 is assigned to class C_2 or vice versa. The probability of this is

Decision boundary – On the left you select class C_1 and on the right class C_2. Based on that you compute the misclassification error.



$$p(mistake) = p(\boldsymbol{x} \in \mathcal{R}_1, C_2) + p(\boldsymbol{x} \in \mathcal{R}_2, C_1) = \underbrace{\int_{\mathcal{R}_1} p(\boldsymbol{x}, C_2)d\boldsymbol{x}}_{\text{Red and green areas}} + \underbrace{\int_{\mathcal{R}_2} p(\boldsymbol{x}, C_1)d\boldsymbol{x}}_{\text{Blue area}}$$

Minimizing the Misclassification Rate

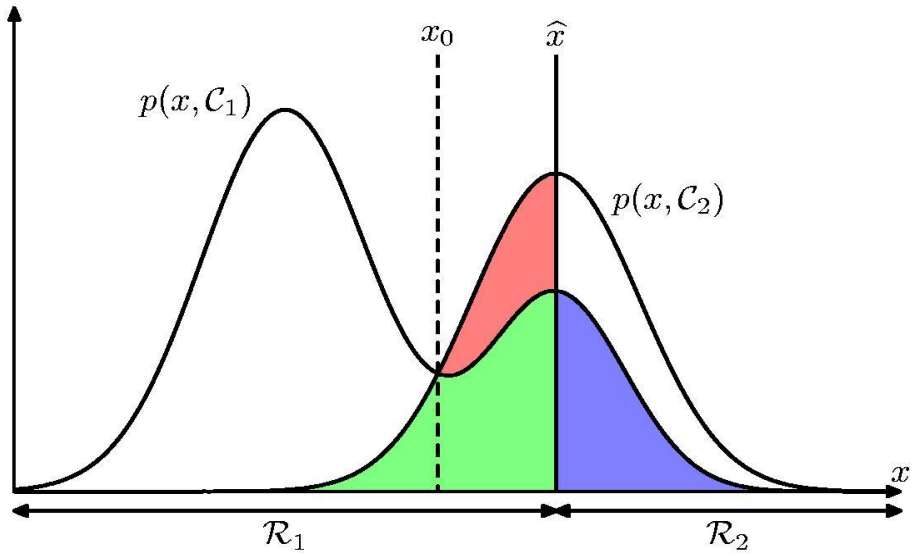


- The optimal selection of the decision boundary corresponds to selecting the class with the highest posterior $p(C_k|\boldsymbol{x})$

Minimizing the Misclassification Rate

- For the more general case of K *classes*, it is slightly easier to maximize the probability of being correct:

$$p(correct) = \sum_{k=1}^{K} p(\boldsymbol{x} \in \mathcal{R}_k, C_k)$$



$$= \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(\boldsymbol{x}, C_k) d\boldsymbol{x} = \sum_{k=1}^{K} \int_{\mathcal{R}_k} p(C_k \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}$$

- This is maximized when each $\boldsymbol{x}$ is assigned to the class for which $p(C_k, \boldsymbol{x})$ is maximum.
- Since $p(\boldsymbol{x})$ is common in all terms, to $\max p(correct)$ we assign each $\boldsymbol{x}$ to the class with $\max p(C_k|\boldsymbol{x})$.

- Note: Note that for 2 classes when posing this as the min of the probability of misclassification, we can write:

$$p(mistake) = \int_{\mathcal{R}_1} p(C_2 \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x} + \int_{\mathcal{R}_2} p(C_1 \mid \boldsymbol{x}) p(\boldsymbol{x}) d\boldsymbol{x}, \text{ where } \mathcal{R}_1 \text{ chosen s.t. } p(C_1 \mid \boldsymbol{x}) > p(C_2 \mid \boldsymbol{x}), etc.$$

$$Using\ a \le b, a, b > 0 \Rightarrow a \le \sqrt{ab},\ we\ can\ show: p(mistake) \le \int_{\mathcal{R}} \sqrt{p(C_1 \mid \boldsymbol{x}) p(C_2 \mid \boldsymbol{x})}\, p(\boldsymbol{x}) d\boldsymbol{x}$$

Minimizing the Expected Loss

- Often our objective is more complex than simply minimizing the number of mis-classifications.
- How do we account that the error of misclassifying a patient with cancer as normal is more serious than classifying a normal patient as having cancer.
- Introduce a *loss (cost) function.* This is a measure of loss incurred in taking any of the available decisions.
- Our goal is then to minimize the total loss incurred.

Loss Matrix

$$
\begin{array}{cc} & \text{Decision} \\ & \begin{array}{cc} \text{cancer} & \text{normal} \end{array} \\ \text{True Class} \begin{array}{c} \text{cancer} \\ \text{normal} \end{array} & \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}
$$

- Loss matrix changes overtime can easily be accounted for.
- Any consistent preferences can be converted to a scalar loss/utility function.

- DeGroot, M. (1970). *Optimal Statistical Decisions*. McGraw-Hill.

Minimizing the Expected Loss

- Suppose that, for a new value of $\boldsymbol{x}$, the true class is C_k and that we assign $\boldsymbol{x}$ to class C_j (where j may or may not be equal to k). In so doing, we incur a loss L_{kj}

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj} \underbrace{p(\mathbf{x}, C_k)}_{\substack{\text{Uncertainy about } \mathbf{x} \\ \text{belonging to class } C_k}} d\mathbf{x} = \sum_j \left(\int_{\mathcal{R}_j} \sum_k L_{kj}\, p(\mathbf{x}, C_k) d\mathbf{x} \right)$$

Average loss wrt the $p(\mathbf{x}, C_k)$ distribution

- Regions $\mathcal{R}_j$ are chosen to minimize $\mathbb{E}[L]$, which implies that for each $\boldsymbol{x}$ we should minimize:

$$\sum_k L_{kj}\, p(\mathbf{x}, C_k)$$

- Assign each $\boldsymbol{x}$ to the class j for which $\sum_k L_{kj}\, p(C_k \mid \mathbf{x})$ is minimum.

Minimizing the Misclassification Rate

$$\mathbb{E}[L] = \sum_k \sum_j \int_{\mathcal{R}_j} L_{kj}\, p(\boldsymbol{x}, C_k) d\boldsymbol{x} = \sum_j \int_{\mathcal{R}_j} \left\{ \sum_k L_{kj}\, p(C_k \mid \boldsymbol{x}) \right\} p(\boldsymbol{x}) d\boldsymbol{x}$$

- The true class here is k and you are assigning it to j
- Regions $\mathcal{R}_j$ are chosen at each $\boldsymbol{x}$ to minimize

$$\sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})$$

- Note: To see the trade-off between L_{kj} *and* $p(C_k)$, for each $\boldsymbol{x}$, we can minimize the following:

$$\frac{1}{p(\boldsymbol{x})} \sum_k \left(L_{kj}\, p(C_k) \right) p(\boldsymbol{x} \mid C_k)$$

Minimizing the Misclassification Rate

- The expected risk is minimized if for each $\boldsymbol{x}$ we choose the class that minimizes $\sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})$
- Let us choose the loss matrix as: $L_{kj} = 1 - \delta_{kj}$
- Then our criterion becomes:

$$\sum_k \left(1 - \delta_{kj}\right) p(C_k \mid \boldsymbol{x}) = \sum_k p(C_k \mid \boldsymbol{x}) - p(C_j \mid \boldsymbol{x}) = 1 - p(C_j \mid \boldsymbol{x})$$

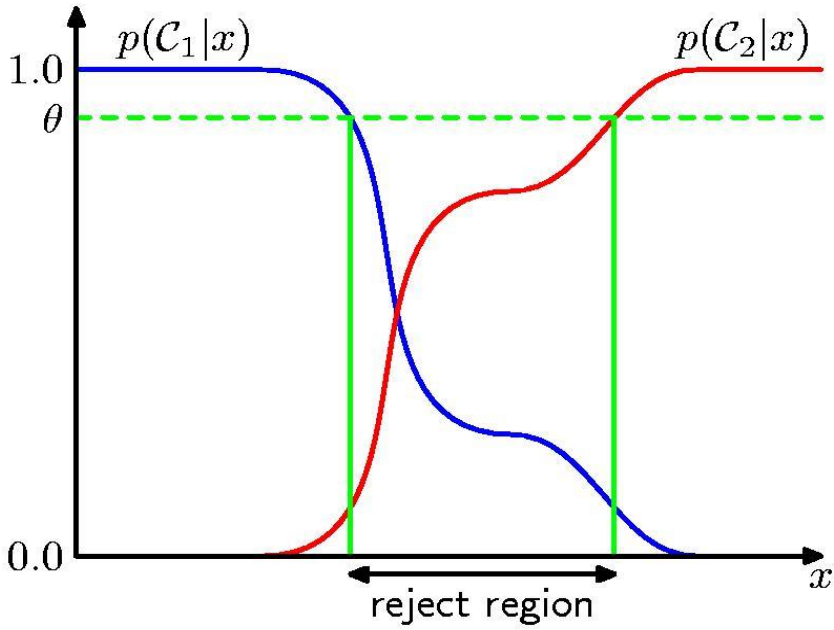
- This is equivalent to choosing j for which the posterior probability $p(C_j \mid \boldsymbol{x})$ is maximized.
- The particular loss matrix assigns a loss of 1 if the example is misclassified and a loss of zero if it is correctly classified – hence minimizing the expected loss will minimize the misclassification rate.

Reject Option

- We should avoid making decisions on difficult cases. This is known as the ***reject option.***

- We introduce a threshold θ and reject those inputs $\boldsymbol{x}$ for which the largest of $p(C_k|\boldsymbol{x})$ is less than or equal to θ.

- For $\theta = 1$ all examples are rejected, whereas for K classes $\theta < 1/K$ ensures that no examples are rejected.
  The fraction of examples that get rejected is controlled by θ.



- We can minimize the expected loss when a reject decision is made.

Loss Function and the Reject Method

- Consider a combination of the loss function approach and the reject method as follows:

$$choose \begin{cases} class\ j, & if \ \min_l \sum_k L_{kl}\, p(C_k \mid \boldsymbol{x}) < \lambda \\ reject, & otherwise \end{cases}, \quad where \quad j = \arg\min_l \underbrace{\sum_k L_{kl}\, p(C_k \mid \boldsymbol{x})}_{Cost\ assigning\ \boldsymbol{x}\ to\ class\ l}$$

- Let us choose the loss matrix as: $L_{kj} = 1 - \delta_{kj}$

$$\min_l \sum_k L_{kl}\, p(C_k \mid \boldsymbol{x}) > \lambda \Rightarrow 1 - \max_l p(C_l \mid \boldsymbol{x}) > \lambda \Rightarrow$$

$$\max_l p(C_l \mid \boldsymbol{x}) < 1 - \lambda$$

- For this choice of the loss matrix, the criterion above (the min assignment acceptance cost $\min_l \sum_k L_{kl}\, p(C_k \mid \boldsymbol{x})$ should be λ) becomes equivalent to the reject criterion for

$$\theta = 1 - \lambda$$

Reject Option in Classifiers

- Consider C classes and you make the decision a_i, $i = 1, \ldots C+1$ where the $C+1$ choice is the reject one.

$$L(a_i \mid y = j) = \begin{cases} 0 & if\ i = j\ and\ i, j = 1, \ldots, C \\ \lambda_r & if\ i = C+1 \\ \lambda_s & otherwise\ (miss - classify) \end{cases}$$

- We choose between rejecting and choosing the most probable class $j_{\max} = \arg\max_j p(y = j \mid x)$. Comparing the risks:

$$\lambda_r \geq \lambda_s \underbrace{\left(1 - p(y = j_{\max} \mid x)\right)}_{\substack{probaility\ that\ j_{\max} \\ was\ the\ wrong\ choice}} \Rightarrow p(y = j_{\max} \mid x) \geq 1 - \frac{\lambda_r}{\lambda_s}$$

- We thus select the most probable class if the above criterion is satisfied, otherwise we reject.

- You can verify that any other choice leads to higher cost:

$$\lambda_s \left(1 - p(y = j \mid x)\right) \geq \lambda_s \left(1 - p(y = j_{\max} \mid x)\right)$$

Summary: Inference and Decision

- We have broken the classification problem down into two separate stages
  - the inference stage in which we use training data to learn a model for $p(C_k|\boldsymbol{x})$, and
  - the subsequent decision stage in which we use these posterior probabilities to make optimal class assignments.

Inference and Decision

- There are three approaches to decision problems:
  - Generative models: Model $p(\boldsymbol{x}, C_k)$ (or $p(\boldsymbol{x}|C_k)$ and the prior $p(C_k)$ for each class) and then normalize to get the posterior.

    The model is called generative since it allows us to generate data in the input space.

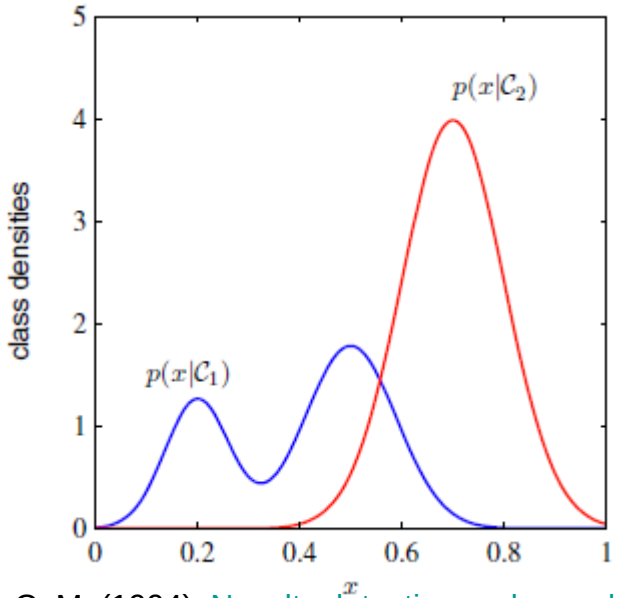
    The class priors can be estimated from the fractions of the training data in each class.
  - Discriminative models: Model directly the posteriors $p(C_k|\boldsymbol{x})$ and then use decision theory to assign to a class a new $\boldsymbol{x}$.

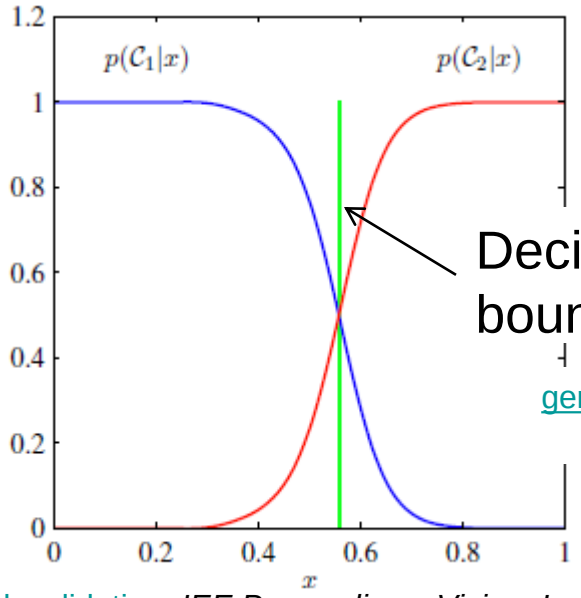
    Models combining generative and discriminative approaches have also been proposed.
  - Discriminant function: Map directly the input $\boldsymbol{x}$ to a class label without any probabilities.

- Jebara, T. (2004). *Machine Learning: Discriminative and Generative*. Kluwer (also here)
- Lasserre, J., C. M. Bishop, and T. Minka (2006). Principled hybrids of generative and discriminative models. In *Proceedings 2006 IEEE Conference on Computer Vision and Pattern Recognition, New York*.

Generative Models

- Generative models are the most expensive.
- Computing the normalization factor $p(\boldsymbol{x})$ allows us to detect outliers (detect data with low probability) $p(\boldsymbol{x}) = \sum_k p(\boldsymbol{x} \mid C_k) p(C_k)$
- Often the class conditional probabilities $p(\boldsymbol{x}|C_k)$ (see the bimodal nature of the blue conditional on the left) have little effect on the needed posteriors as the plot below from Bishop's PRML shows:





generativeVsDiscrim from PMTK

- Bishop, C. M. (1994). Novelty detection and neural network validation. *IEE Proceedings: Vision, Image and Signal Processing* **141**(4), 217–222. Special issue on applications of neural networks.
- Tarassenko, L. (1995). Novelty detection for the identification of masses in mamograms. In *Proceedings Fourth IEE International Conference on Artificial Neural Networks*, Volume 4, pp. 442–447. IEE.

Merit for Computing the Posteriors

- There are many powerful reasons for wanting to compute the posterior probabilities $p(C_k|\boldsymbol{x})$, even if we subsequently use them to make decisions.

- These include:

  - Minimizing risk (loss matrix may change over time, easy to revise the min risk decision criterion $\sum_k L_{kj}\, p(C_k \mid \boldsymbol{x})$)
  - Reject option (easy to minimize the misclassification rate with given posterior probabilities)
  - Unbalanced class priors
  - Combining models
  - Other

Unbalanced Class Priors

- In medical screening applications, cancer is very rare.
- We need to use balanced data to train models (e.g. 5000 cancer cases and 5000 normal cases), then use Bayes' rule to compute the posterior probabilities for the artificially balanced data set.

$$p(C_k \mid \boldsymbol{x}) = \frac{p(\boldsymbol{x} \mid C_k)}{p(\boldsymbol{x})} p(C_k)$$

- Since the prior probabilities can be interpreted as the fractions of points in each class, we finally write:

$$\textit{Updated Posterior Probability} \propto p(C_k \mid \boldsymbol{x}) \frac{\textit{Class fraction in the population where we want to apply the model}}{\textit{Class fractions in the data set used}}$$

- Finally, the computed posteriors needs to be normalized.

Combining Models

- In a medical diagnosis problem, we may have hete-rogeneous information e.g. blood tests & X −ray images.
- Build one system to interpret the X −ray images and a different one to interpret the blood data.

  Assuming these two models give $p(C_k|\boldsymbol{x}_I), p(C_k|\boldsymbol{x}_B)$, we can combine the outputs systematically.
- One simple way to do this is by assuming that for each class separately, the distributions of inputs for the X −ray images, denoted by $\boldsymbol{x}_I$, and the blood data, denoted by $\boldsymbol{x}_B$, are independent.

$$p(\boldsymbol{x}_I, \boldsymbol{x}_B \mid C_k) \propto p(\boldsymbol{x}_I \mid C_k)\, p(\boldsymbol{x}_B \mid C_k)$$

Combining Models: Naïve Bayes Model

$$p(\boldsymbol{x}_I, \boldsymbol{x}_B \mid C_k) \propto p(\boldsymbol{x}_I \mid C_k)\, p(\boldsymbol{x}_B \mid C_k)$$

- This is an example of conditional independence (naïve Bayes model).
- Given both data, the posterior probability is then given by

$$\begin{aligned} p(C_k \mid \boldsymbol{x}_I, \boldsymbol{x}_B) &= p(\boldsymbol{x}_I, \boldsymbol{x}_B \mid C_k)\, p(C_k) \\ &\propto p(\boldsymbol{x}_I \mid C_k)\, p(\boldsymbol{x}_B \mid C_k)\, p(C_k) \\ &\propto \frac{p(C_k \mid \boldsymbol{x}_I)\, p(C_k \mid \boldsymbol{x}_B)}{p(C_k)} \end{aligned}$$

- Compute the needed $p(C_k)$ from the fractions of data points in each class.
- Finally, normalize the resulting posterior probabilities.

False Positive vs. False Negative

- Binary decision problems: hypothesis testing, two-class classification, object/event detection, etc.

- There are two types of error:

  - a false positive (aka false alarm), which arises when we estimate $\hat{y} = 1$ but the truth is $y = 0$; or

  - a false negative (aka missed detection), which arises when we estimate $\hat{y} = 0$ but the truth is $y = 1$.

- We represent this with a loss matrix.

Loss Matrix

$$
\begin{array}{c} \\ True\ Class \end{array}
\begin{array}{c} \\ y=1 \\ y=0 \end{array}
\begin{array}{c} Decision \\ \hat{y}=1 \quad \hat{y}=0 \\ \begin{pmatrix} 0 & L_{FN} \\ L_{FP} & 0 \end{pmatrix} \end{array}
$$

- L_{FN} is the cost of a false negative, and L_{FP} is the cost of a false positive.
- *The posterior expected loss for the two possible actions is given next*.

False Positive vs. False Negative

Posterior Expected Losses

$$\rho(\hat{y}=0|\boldsymbol{x}) = L_{FN}p(y=1|\boldsymbol{x})$$
$$\rho(\hat{y}=1|\boldsymbol{x}) = L_{FP}p(y=0|\boldsymbol{x})$$

- One should pick the class $\hat{y}=1$ iff $\rho(\hat{y}=0|\boldsymbol{x}) > \rho(\hat{y}=1|\boldsymbol{x}) \Rightarrow$

$$\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})} > \frac{L_{FP}}{L_{FN}}, \Rightarrow \frac{p(\boldsymbol{x}|y=1)}{p(\boldsymbol{x}|y=0)} > \frac{L_{FP}}{L_{FN}}\frac{p(y=0)}{p(y=1)} = \frac{L_{FP}}{L_{FN}}\frac{1-\pi}{\pi}$$

- The optimal decision rule is always a likelihood ratio.

- If $L_{FN} = cL_{FP}$, it is easy to show that we should pick $\hat{y}=1$ iff

$$\frac{p(y=1|\boldsymbol{x})}{p(y=0|\boldsymbol{x})} > \frac{1}{c} \text{ or } p(y=1|\boldsymbol{x}) > \tau \equiv \frac{1}{1+c}$$

- E.g., if $c = 2$, then we use a decision threshold of $1/3$ before declaring a positive.

- Muller, P., G. Parmigiani, C. Robert, and J. Rousseau (2004). Optimal sample size for multiple testing: the case of gene expression microarrays. *J. of the Am. Stat. Assoc. 99*, 990–1001.

False Positives Vs. False Negatives

- *False positive (FP):* Predict class 1 when truth is class 0
- *False negative (FN):* Predict class 0 when truth is class 1
- *True positive (TP):* Predict class 1 when truth is class 1
- *True negative (TN):* Predict class 0 when truth is class 0

$\hat{N}_+$ is the "called" number of positives,
$\hat{N}_-$ is the "called" number of negatives.

		Truth		
		1	0	Σ
Estimate	1	TP	FP	$\hat{N}_+ = TP + FP$
	0	FN	TN	$\hat{N}_- = FN + TN$
	Σ	$N_+ = TP + FN$	$N_- = FP + TN$	$N = TP + FP + FN + TN$

N_+ is the true number of positives,
N_- is the true number of negatives

- An introduction to ROC analysis, Tom Fawcett, 2006
- T. Fawcett, ROC Graphs: Notes and Practical Considerations for Researchers, 2004
- Additional intro resources can be found here and here

Confusion Matrix

- Consider a binary decision problem, e.g. classification, or object detection.
- We have a labeled data set, $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}$. Let

$$\delta(\boldsymbol{x}) = \mathbb{I}(f(\boldsymbol{x}) > \tau)$$

be our decision rule, where $f(\boldsymbol{x})$ is a measure of confidence that $y = 1$ (this should be monotonically related to $p(y = 1|\boldsymbol{x})$, but does not need to be a probability), and τ is a threshold parameter.

- For each value of τ , we apply our decision rule and count the number of true positives, false positives, true negatives, and false negatives that occur. This table of errors is called a **confusion matrix**.

$\hat{N}_+$ is the "called" number of positives,
$\hat{N}_-$ is the "called" number of negatives.

		Truth		
		1	0	Σ
Estimate	1	TP	FP	$\hat{N}_+ = TP + FP$
	0	FN	TN	$\hat{N}_- = FN + TN$
	Σ	$N_+ = TP + FN$	$N_- = FP + TN$	$N = TP + FP + FN + TN$

N_+ is the true number of positives,
N_- is the true number of negatives

Receiver Operating Characteristic (ROC) Curves

		Truth		
		1	0	Σ
Estimate	1	TP	FP	$\hat{N}_+ = TP + FP$
	0	FN	TN	$\hat{N}_- = FN + TN$
	Σ	$N_+ = TP + FN$	$N_- = FP + TN$	$N = TP + FP + FN + TN$

- From this table, we can compute the **true positive rate** (TPR), also known as the **sensitivity**, **recall** or **hit rate**,

$$TPR = TP/N^+ \approx p(\hat{y} = 1 | y = 1)$$

N_+ is the true number of positives

- We can also compute the **false positive rate** (FPR), also called the **false alarm rate**, or the **type I error rate**,

$$FPR = FP/N^- \approx p(\hat{y} = 1 | y = 0)$$

N_- is the true number of negatives

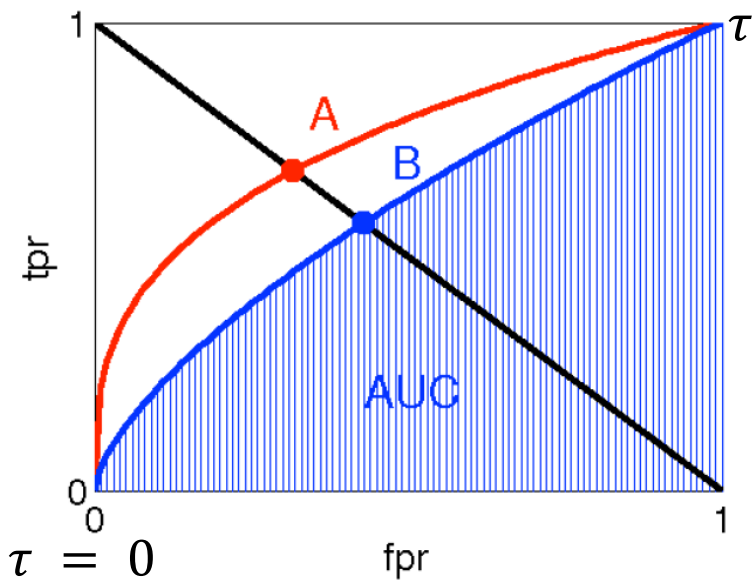
	$y = 1$	$y = 0$
$\hat{y} = 1$	TP/N_+=TPR=sensitivity=recall	FP/N_-=FPR=type I
$\hat{y} = 0$	FN/N_+=FNR=miss rate=type II	TN/N_-=TNR=specifity

- **Receiver operating characteristic (ROC) curve**: Run the classifier for various threshholds τ, and plot TPR vs FPR as a function of τ.

Receiver Operating Characteristic (ROC) Curves

- $FPR = 0, TPR = 0$ is obtained by $\tau = 1$ and thus classifying everything as negative.
- $FPR = 1, TPR = 1$ is obtained by $\tau = 0$ and thus classifying everything as positive.
- If a system is performing at chance level, we can achieve any point on the line $TPR = FPR$ by choosing an appropriate threshold.
- A system that perfectly separates the positives from negatives has a τ that can achieve $FPR = 0, TPR = 1$; by varying the threshold such a system will "hug" the left axis and then the top axis.

A better than B

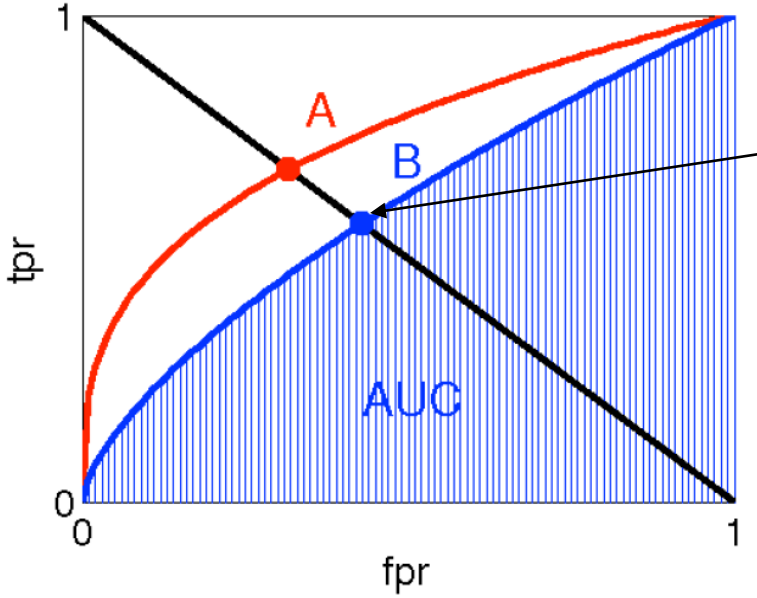


Declare $\boldsymbol{x}_n$ to be a positive if $p(y = 1|\boldsymbol{x}_n) > \tau$, otherwise declare it to be negative ($y = 0$)

$$\hat{y}_n = 1 \leftrightarrow p(y = 1|\boldsymbol{x}_n) > \tau$$

- An ROC curve is used to visualize the performance of a binary classifier.
- AUC summarizes its performance in a single number.

Area Under the Curve / Equal Error Rate



$$FPR = FNR = 1 - TPR$$

$$FPR = FP/N^- \approx p(\hat{y} = 1|y = 0)$$
$$TPR = TP/N^+ \approx p(\hat{y} = 1|y = 1)$$
$$FNR = FN/N^+ \approx p(\hat{y} = 0|y = 1)$$

- The quality of a ROC curve is often summarized using the **area under the curve**. Higher AUC scores are better; the max is 1.

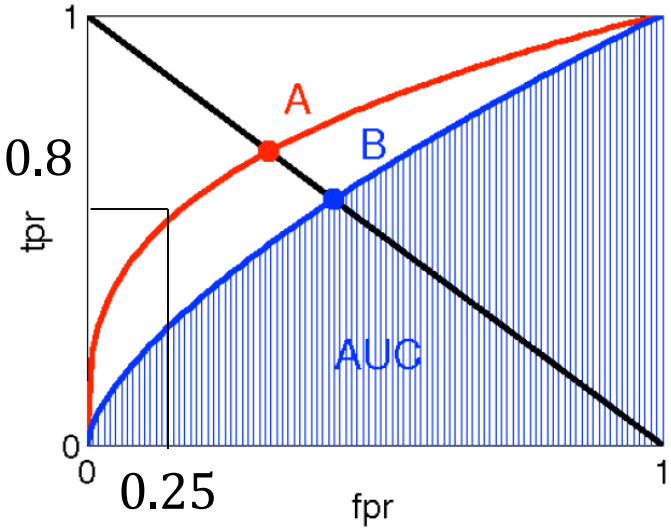
- Another summary statistic that is used is the **equal error rate** or **EER**, also called the **cross over rate**, defined as the value which satisfies $FPR = FNR = 1 - TPR$, we can compute the EER by drawing a line from the top left to the bottom right and seeing where it intersects the ROC curve (see points A and B). Lower EER scores are better; the minimum is obviously 0.

ROC Curves - Example

i	y_i	$p(y_i = 1\|\boldsymbol{x}_i)$	$\hat{y}_i(\tau = 0)$	$\hat{y}_i(\tau = 0.5)$	$\hat{y}_i(\tau = 1)$
1	1	0.9	1	1	0
2	1	0.8	1	1	0
3	1	0.7	1	1	0
4	1	0.6	1	1	0
5	1	**0.2**	1	0	0
6	0	**0.6**	1	1	0
7	0	0.3	1	0	0
8	0	0.2	1	0	0
9	0	0.1	1	0	0

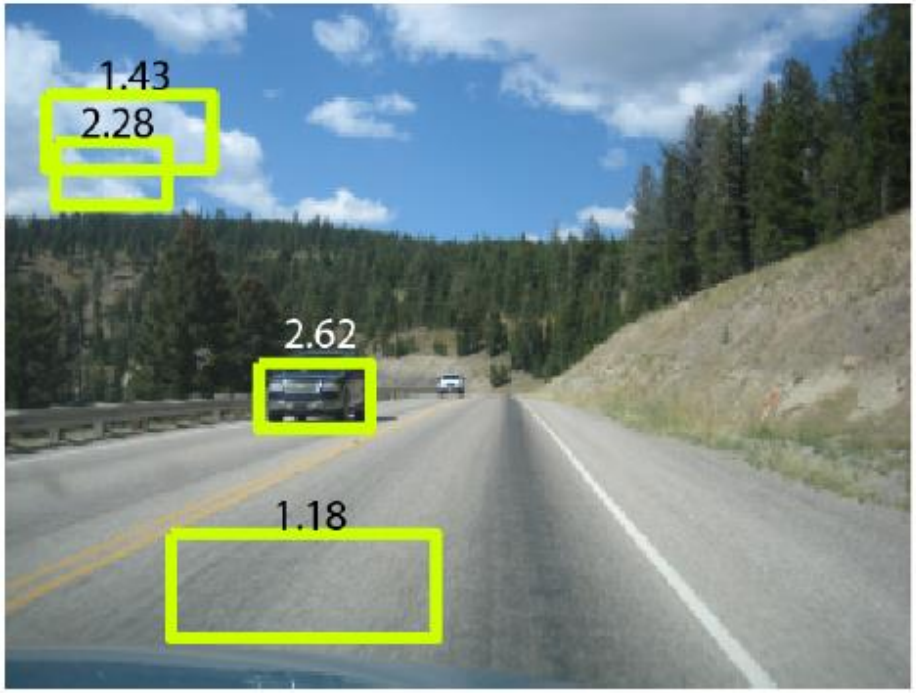
$$TPR = \frac{4}{5} = 0.8,$$

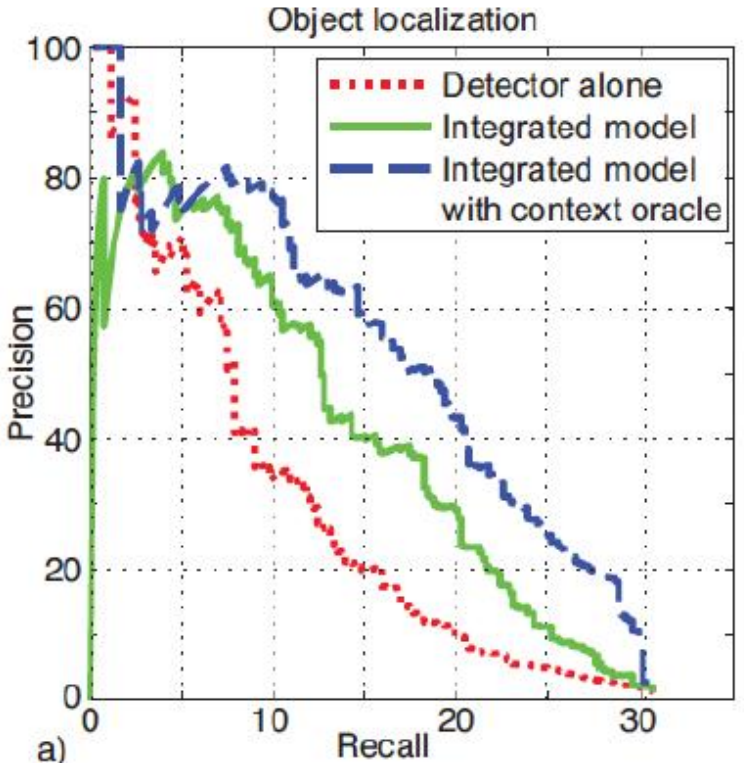
$$FPR = \frac{1}{4} = 0.25$$



Precision Recall Curves

- The number of *negative* examples may not be well defined.
- How many windows not containing a car are there in an image?

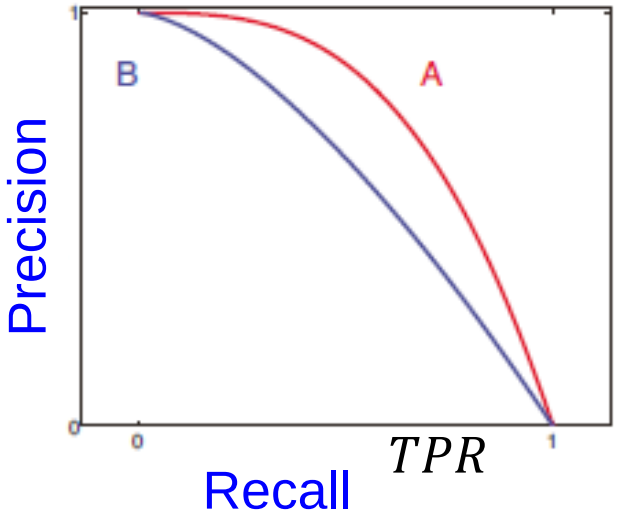




- **Recognizing and Learning Object Categories**, at ICCV 2005 Li Fei-Fei (Stanford), Rob Fergus (NYU), Antonio Torralba (MIT)

Precision Recall Curves

- Often, the notion of negative is not well-defined.
- E.g., when detecting objects in images, if the detector works by classifying patches, then the number of patches examined (and hence the number of true negatives) is a parameter of the algorithm, not part of the problem definition.



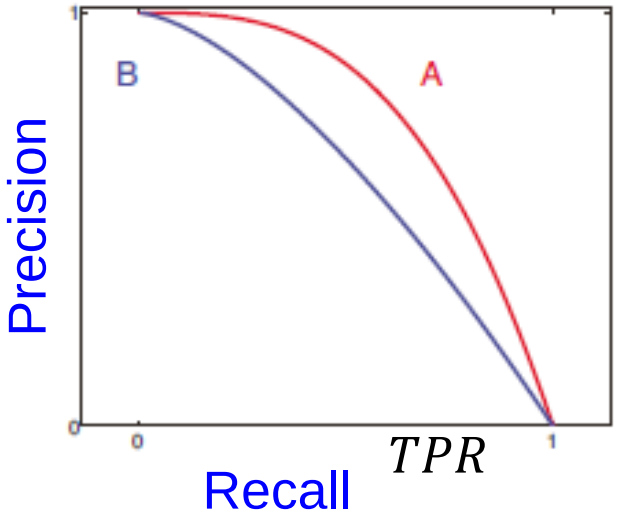
- *So we would like to use a measure that only talks about positives.*

$$R = \frac{TP}{N_+} = p(\hat{y} = 1 \mid y = 1) = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$$

$$P = \frac{TP}{N_+} = p(y = 1 \mid \hat{y} = 1) = \frac{\sum_i \hat{y}_i y_i}{\sum_i \hat{y}_i}$$

Precision Recall Curves

- When trying to detect a rare event (e.g. retrieving a relevant document), the number of negatives is very large. Comparing $TPR = \frac{TP}{N_+}$ to $FPR = \frac{FP}{N_-}$ is not informative, since the FPR is very small.



- All the action in the ROC curve occurs on the extreme left. In such cases, it is common to *plot the TPR versus the number of false positives, rather than vs the false positive rate*.

$$R = \frac{TP}{N_+} = p(\hat{y} = 1 \mid y = 1) = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$$

$$P = \frac{TP}{N_+} = p(y = 1 \mid \hat{y} = 1) = \frac{\sum_i \hat{y}_i y_i}{\sum_i \hat{y}_i}$$

Recall = of those that exist, how many did you find?

Precision = of those that you found, how many were correct?

Precision Recall Curves

		Truth		
		1	0	Σ
Estimate	1	TP	FP	$\hat{N}_+ = TP + FP$
	0	FN	TN	$\hat{N}_- = FN + TN$
	Σ	$N_+ = TP + FN$	$N_- = FP + TN$	$N = TP + FP + FN + TN$

	$y = 1$	$y = 0$
$\hat{y} = 1$	TP/N_+=TPR=sensitivity=recall	FP/N_-=FPR=type I
$\hat{y} = 0$	FN/N_+=FNR=miss rate=type II	TN/N_-=TNR=specifity

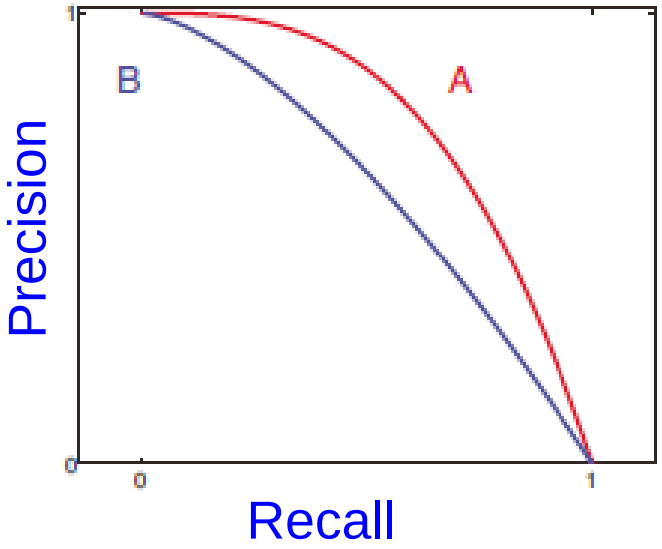
- The **precision** *P* and **recall** *R* are defined as

$$P = \frac{TP}{\hat{N}_+} = p(y = 1|\hat{y} = 1) = \frac{\sum_i \hat{y}_i y_i}{\sum_i \hat{y}_i}, \quad R = \frac{TP}{N_+} = p(\hat{y} = 1|y = 1) = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$$

- Precision: What fraction of our detections are actually positive.
- Recall: What fractions of the positives we actually observed.

Precision Recall Curves

$$P = \frac{TP}{N_+} = p(y=1 \mid \hat{y}=1) = \frac{\sum_i \hat{y}_i y_i}{\sum_i \hat{y}_i}, \quad R = \frac{TP}{N_+} = p(\hat{y}=1 \mid y=1) = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$$



PRhand from Kevin Murphys' PMTK

- *Precision measures what fraction of our detections are actually positive, and recall measures what fraction of the positives we actually detected*.
  Here $\hat{y}_i = \{0,1\}$ *and* $y_i = \{0,1\}$ are the predicted labels, and the true labels.
- *A **precision recall curve** is a plot of precision vs recall as we vary the threshold* τ. Hugging the top right is the best one can do.
- This curve can be presented using *the mean precision averaging over recall values*. This approximates the area under the curve.
- Alternatively, one can *quote the precision for a fixed recall level*, such as the precision of the first $K = 10$ entities recalled. This is called the average precision at *K* score (used in evaluating information retrieval systems).

F Scores

- For a fixed threshold, one can compute a single precision and recall value. These are combined into a single statistic the ***F score***, *or* ***F1 score***, *which is the harmonic mean of* P *and* R: $F_1 = \frac{2PR}{R+P}$
- Using $P = \frac{TP}{N_+} = p(y=1|\hat{y}=1) = \frac{\sum_i \hat{y}_i y_i}{\sum_i \hat{y}_i}, \quad R = \frac{TP}{N_+} = p(\hat{y}=1|y=1) = \frac{\sum_i y_i \hat{y}_i}{\sum_i y_i}$

  we can write this as $F_1 = \frac{2\sum_i y_i \hat{y}_i}{\sum_i y_i + \sum_i \hat{y}_i}$
- This is a widely used measure in information retrieval systems.
- Let us see why we use the harmonic mean instead of the arithmetic mean, $(P + R)/2$.
- Suppose we recall all entries, so $R = 1$. Suppose the precision is low, $P = 10^{-4}$. The arithmetic mean of P and R is $(P + R)/2 = (10^{-4} + 1)/2 \approx 50\%$. By contrast, the harmonic mean is only $2 \times 10^{-4} \times 1/(1 + 10^{-4}) \approx 0.2\%$.

F Scores – A Word of Caution

- Consider binary classifiers A, B, C

y		A	.	B	.	C	.
		1	0	1	0	1	0
$\hat{y}$	1	0.9	0.1	0.8	0	0.78	0
	0	0	0	0.1	0.1	0.12	0.1

- Clearly A is useless, since it always predicts label 1, regardless of the input.

- B is slightly better than C (less probability mass wasted on the off diagonal entries).

- Yet here are the performance metrics.

Metric	A	B	C
Accuracy	0.9	0.9	0.88
Precision	0.9	1.0	1.0
Recall	1.0	0.888	0.8667
F-score	0.947	0.941	0.9286

Mutual Information

- The MI between estimated and true labels is

$$I(\hat{Y}, Y) = \sum_{\hat{y}=0}^{1} \sum_{y=0}^{1} p(\hat{y}, y) log \frac{p(\hat{y}, y)}{p(\hat{y})p(y)}$$

- This gives the intuitively correct rankings B>C>A

Metric	A	B	C
Accuracy	0.9	0.9	0.88
Precision	0.9	1.0	1.0
Recall	1.0	0.888	0.8667
F-score	0.947	0.941	0.9286
Mutual information	0	0.1865	0.1735

Micro- and Macro-Averaged F Scores

- In the multi-class case (e.g. document classification problems), there are two ways to generalize $F1$ scores.

- ***Macro-averaged F1*** *defined as* $\sum_{c=1}^{C} F_1(c)/C$ *where* $F_1(c)$ *is the* F_1 *score obtained on the task of distinguishing class* c *from all the others.*

- ***Micro-averaged F1*** *defined as the* $F1$ *score where we pool all the counts from each class's contingency table*.

- Example: the precision of class 1 is 0.5, and of class 2 is 0.9. The macro-averaged precision is 0.7, whereas the micro-averaged precision is $100/120 = 0.83$. The latter is closer to the precision of class 2 than to the precision of class 1, since class 2 is five times larger than class 1. *To give equal weight to each class, use macro-averaging.*

Class 1		
	$y=1$	$y=0$
$\hat{y}=1$	10	10
$\hat{y}=0$	10	970

Class 2		
	$y=1$	$y=0$
$\hat{y}=1$	90	10
$\hat{y}=0$	10	890

Pooled		
	$y=1$	$y=0$
$\hat{y}=1$	100	20
$\hat{y}=0$	20	1860

False Discovery Rates

- Suppose we are trying to discover a rare phenomenon using a high throughput measurement device.
- We need to make many binary decisions of the form $p(y_i = 1|\mathcal{D}) > \tau$, where $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N$ and N may be large.
- This is called ***multiple hypothesis testing***. We are classifying y_i based on all data not just based on $\boldsymbol{x}_i$. This way we hope to do better than in a series of individual classification problems.
- To set the threshold τ, *minimize the expected number of false positives*. In the Bayesian approach, this can be computed as follows:

$$FD(\tau, \mathcal{D}) = \sum_i (1 - p_i)\mathbb{I}(p_i > \tau), \; p_i = p(y_i = 1|\mathcal{D})$$

- $p_i = p(y_i = 1|\mathcal{D})$ is the belief that this object exhibits the phenomenon in question so $1 - p_i$ is the probability of error.

False Discovery Rates

- We then define *the posterior expected* ***false discovery rate*** as follows:

$$FDR(\tau, \mathcal{D}) = FD\,(\tau, \mathcal{D})/N\,(\tau, \mathcal{D}),\ N(\tau, \mathcal{D}) = \sum_i \mathbb{I}(p_i > \tau)$$

- $\mathbb{I}(p_i > \tau)$ is the number of discovered items. Given a desired FDR tolerance, say $\alpha = 0.05$, one can then adapt τ to achieve this.

- This is the **direct posterior probability approach** to controlling the FDR.

- Estimate the p_i's jointly using a hierarchical Bayesian model. This allows the pooling of statistical strength, and thus lower FDR.

- Newton, M., D. Noueiry, D. Sarkar, and P. Ahlquist (2004). Detecting differential gene expression with a semiparametric hierarchical mixture method. *Biostatistics 5*, 155– 176.
- Muller, P., G. Parmigiani, C. Robert, and J. Rousseau (2004). Optimal sample size for multiple testing: the case of gene expression microarrays. *J. of the Am. Stat. Assoc. 9*, 990–1001.
- Berry, D. and Y. Hochberg (1999). Bayesian perspectives on multiple comparisons. *J. Statist. Planning and Inference 82*, 215–227.

Contextual Bandits

- ***One-armed bandit:*** colloquial term for a slot machine.

- ***Multi-armed bandit****:* Choosing from K machines. Let r_k be the reward pulling the arm of the K machine. We device an optimal policy using

$$p\ r_{1:K} \mid \mathcal{D} = \prod_k p\ r_k \mid \mathcal{D}$$

- This is compiled into a series of *Gittins Indices* optimally solving the *exploration-exploitation* tradeoff: *how many times we should try each action before deciding going with the winner*?

- Gittins, J. (1989). *Multi-armed Bandit Allocation Indices*. Wiley.
- Sarkar, J. (1991). One-armed bandit problems with covariates. *The Annals of Statistics 19*(4), 1978–2002.
- Scott, S. (2010). A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry 26*, 639–658.
- Li, L., W. Chu, J. Langford, and X. Wang (2011). Unbiased offline evaluation of contextual-bandit based news article recommendation algorithms. In *WSDM*.

Contextual Bandits

- ***Contextual Bandid:*** Each arm and the player, has an associated feature vector; let all these features $\boldsymbol{x}$. E.g. the "arms" represent news articles which we want to show to the user, and the features represent properties of the articles (bag of words) and of the user (demographics).
- Assume a linear model for reward, $r_k = \boldsymbol{\theta}_k^T \boldsymbol{x}$, and maintain a distribution over the parameters of each arm, $p(\boldsymbol{\theta}_k|\mathcal{D})$, where
  - $\mathcal{D}$ is a series of tuples of the form $(a, \boldsymbol{x}, r)$, which specifies
  - which arm a was pulled,
  - what its features $\boldsymbol{x}$ were, and
  - what the resulting outcome was (e.g., $r = 1$ if the user clicked on the ad, and $r = 0$ otherwise).
- One can compute $p(\boldsymbol{\theta}_k|\mathcal{D})$ using linear or logistic regression.

- Gittins, J. (1989). *Multi-armed Bandit Allocation Indices*. Wiley.
- Sarkar, J. (1991). One-armed bandit problems with covariates. *The Annals of Statistics 19*(4), 1978–2002.
- Scott, S. (2010). A modern Bayesian look at the multi-armed bandit. *Applied Stochastic Models in Business and Industry 26*, 639–658.
- Li, L., W. Chu, J. Langford, and X. Wang (2011). Unbiased offline evaluation of contextual-bandit based news article recommendation algorithms. In *WSDM*.

Contextual Bandits: Upper Confidence Bound

- Given $p(\boldsymbol{\theta}_k|\mathcal{D})$, we must decide what action to take. One common heuristic, UCB (upper confidence bound) is to take the action which maximizes

$$k^* = \arg\max_{k=1:K} \mu_k + \lambda\sigma_k, \mu_k = \mathbb{E}\ r_k \mid \mathcal{D}\ , \sigma_k = \text{var}\ r_k \mid \mathcal{D}$$

- *Here* λ is a tuning parameter that trades off exploration and exploitation.

- By intuition we should pick actions which we believe are good (μ_k is large), and/ or actions which we are uncertain (σ_k is large).

Contextual Bandits: Thompson Sampling

- In a simpler method (Thompson sampling), we *pick action k with a probability that is equal to its probability of being the optimal action:*

$$p_k = \int \mathbb{I}\ \mathbb{E}\ r \mid k, \mathbf{x}, \theta = \max_{a'} \mathbb{E}\ r \mid a', \mathbf{x}, \theta \quad p\ \theta \mid \mathcal{D}\ d\theta$$

- Effectively we find the action that maximizes the reward for a given values of the parameters and then average over the parameters.

- We can approximate this by drawing a single sample from the posterior, $\boldsymbol{\theta}^t \sim p(\boldsymbol{\theta}|\mathcal{D})$, and then choosing

$$k^* = \arg\max_k \mathbb{E}\left[r \mid k, \mathbf{x}, \theta^t\right]$$

- This has been shown to work quite well.

- Chapelle, O. and L. Li (2011). An empirical evaluation of Thompson sampling. In *NIPS*.